Pulling Open Doors and Drawers: Coordinating an Omni-directional Base and a Compliant Arm with Equilibrium Point Control

Advait Jain[1] and Charles C. Kemp[2]

Abstract— **Previously, we have presented an implementation of impedance control inspired by the Equilibrium Point Hypothesis that we refer to as *equilibrium point control* (EPC). We have demonstrated that EPC can enable a robot in a fixed position to robustly pull open a variety of doors and drawers, and infer their kinematics without detailed prior models.**

In this paper, we extend this framework to support autonomous motion of the robot's omni-directional base both before and during pulling. With our new methods, we show that the robot can autonomously approach and open doors and drawers for which only the location and orientation of the handle have been provided. We also demonstrate that EPC can coordinate the movement of the robot's omni-directional base and compliant arm while pulling open a door or drawer, which leads to significantly improved performance.

Through 40 trials with 10 different doors and drawers, we empirically demonstrated the robustness of the system. The robot succeeded in 37 out of 40 trials, and had no more than a single failure for any particular door or drawer.

I. Introduction

A large variety of doors and drawers can be found within human environments. Operating these mechanisms plays a role in many daily activities, such as moving within an environment or retrieving an object that has been stored. Being able to operate these same mechanisms would help service robots assist with similar activities.

Within this paper, we present methods that enable a robot to autonomously approach and pull open a variety of doors and drawers for which only the location and orientation of the handle have been provided. While pulling on the handle, the robot haptically infers the mechanism's kinematics in order to adapt the motion of its base and arm. We empirically demonstrate that the system is robust to common forms of task variation, including variation in the mechanism being operated (tested with 7 doors and 3 drawers), and variation in the pose of the robot's base with respect to the handle.

There are two main contributions of this paper. First, we extend our previous work [1] to enable the robot to autonomously approach and open mechanisms for which only the location and orientation of the handle have been provided. Second, we generalize our previous implementation of equilibrium point control to allow the robot to move its omni-directional base while simultaneously pulling things open. We show that this substantially improves the performance of the robot in this task. Specifically, it allows the robot to open doors over greater angular ranges and drawers over greater distances than before, as shown in Figure 1. In essence, this increases the effective workspace of the manipulator.

Fig. 1. Images showing the robot after it has operated two out of the 10 different mechanisms used in the experiments of Section IX.

In spite of the advantages of moving the base in conjunction with the arm, the majority of research to date on manipulation within human environments holds the base stationary while arm(s) manipulate the world [2], [3]. This decision simplifies many aspects of control, but sacrifices the performance of the mobile manipulator.

Fortunately, the improved performance of our robot comes with only a modest increase in the complexity of the robot's control. As in our previous work, our approach uses a straightforward form of impedance control, which we call *equilibrium point control* (EPC) [1]. As we describe in detail later, EPC can be performed with respect to a frame of reference attached to the environment, and thereby coordinate the motion of the robot's base and arm.

Furthermore, we demonstrate that the motion of the base and the arm only need to be loosely coupled for success in this task. A control loop written in Python, running at approximately 10Hz is sufficient to coordinate the motion of the mobile base and the robot's arm. This makes it easier to use sensors like cameras in the control loop since we do not have strict timing constraints. For example, to estimate the motion of the mobile base, we compute visual odometry using a floor-facing camera at around 10Hz.

A. Low-Impedance Manipulation

Researchers have made compelling arguments for the benefits of robots with low mechanical impedance [4], [5]. As has often been noted, these arguments are particularly relevant for manipulation within human environments, since we expect the robot's perception of the world and predictions about the world to have high uncertainty. At minimum, low-impedance manipulation reduces the forces and moments resulting from contact, and thus reduces the risk of damage to the robot, the environment, and nearby people.

[1,2]Healthcare Robotics Laboratory, Georgia Tech
[1]advait@cc.gatech.edu [2]charlie.kemp@bme.gatech.edu

In addition, we believe that low-impedance manipulation can improve the performance of robots with respect to common tasks within human environments. Although low impedance manipulation often results in poorer performance when moving the end effector through a pre-defined trajectory [6], we believe this type of evaluation fails to capture the challenges of real-world manipulation in human environments.

The common task we address in this paper (opening novel doors and drawers) illustrates a variety of forms of real-world task variation and uncertainty that a service robot is likely to encounter. Our system makes use of low mechanical impedance to robustly operate given this variation and uncertainty. For example, our system requires that the robot be able to haptically explore the environment and make unexpected contact with rigid components of the environment (e.g. when haptically finding and hooking onto the handle). Likewise, our system requires that the robot accommodate unexpected displacements (e.g. the handle's trajectory and uncompensated motion of the arm when the base moves). At their core, these examples require that the world be able to move the robot's arm, and do so without generating large forces and torques. As such, we believe that low-impedance manipulation is well-matched to the task.

Our robot's arm has low mechanical impedance at all links. A service robot in daily operation may not be able to restrict its contact with the world to its end effector. We believe there is value in using robots and control schemes that provide low mechanical impedance for contact at any point along the arm.

Within our research, the end effector stiffness of the manipulator of our robot is relatively low when compared to other impedance controlled arms. For example, it is lower by around a factor of five compared to work on door opening with Cartesian space impedance control using the DLR-Lightweight-Robot-II [7], [8]. It also uses joint stiffnesses that are comparable to stiffness estimates for joints in the human arm during planar manipulation [9], [10].

B. The Equilibrium Point Hypothesis

Previously, we have presented an implementation of impedance control inspired by the Equilibrium Point Hypothesis (EPH) that we refer to as *equilibrium point control* (EPC) [7], [1]. The EPH has a long history that originates in biomechanical models of the spring-like properties of neuromuscular systems [11]. It is a well-known hypothesis about biological motor control, which posits that motion is controlled by adjusting the equilibrium point of a biomechanical system over time [11]. These sequences of equilibrium points are sometimes referred to as virtual trajectories [12]. In this context, the equilibrium point refers to the configuration to which the mechanical system would settle in the absence of externally applied forces other than gravity. The EPH has often been presented as a model of biological control that does not require explicit compensation for dynamics. Long-standing debates continue about whether or not the EPH is true for human motor control [13], [14].

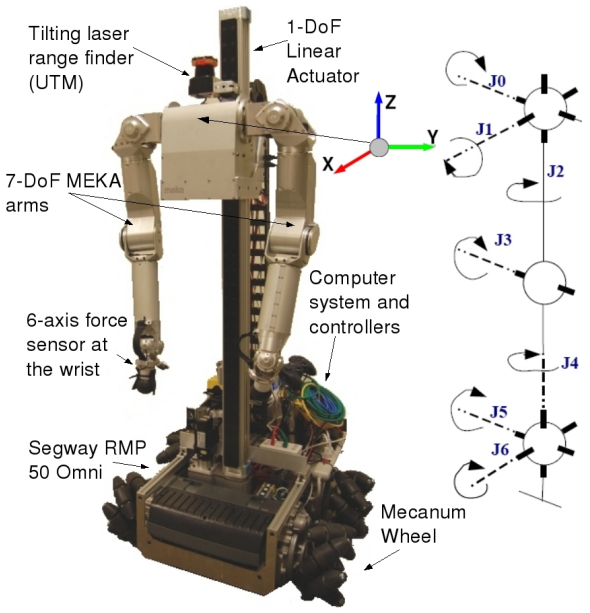

Fig. 2. **Left:** The mobile manipulator used in this paper and the coordinate frame attached to the torso. **Right:** The orientation of the 7 joint axes of each arm (copied with permission from MEKA Robotics datasheets).

C. Equilibrium Point Control

For equilibrium point control (EPC), the motion of the robot's arm is commanded by adjusting the position of a Cartesian-space equilibrium point (CEP) that denotes where the robot's end effector would settle in the absence of externally applied forces other than gravity. For our implementation, this is achieved through the use of virtual visco-elastic springs at the robot's joints along with gravity compensation. For any commanded CEP, we find an associated joint space equilibrium point (JEP) that defines the equilibrium settings for the virtual springs that would result in the robot's end effector settling at the CEP. The robot can also adjust the stiffnesses of these virtual springs. Unlike some approaches to force control and impedance control, we do not explicitly model the dynamics of the arm nor the impedance at the end effector [8]. We also do not use inverse dynamics [15]. As such, equilibrium point control is relatively simple to use.

Previous robotics research has looked at similar robotic control strategies in simulation [16], in freespace motions [17], in legged locomotion [18], in rhythmic manipulation from a fixed based [19], and in the design and control of compliant actuators [20], [21]. However, few researchers have looked at this form of control in the context of task-oriented mobile manipulation. Coupled with our robot's low mechanical impedance, we have found EPC to be easy to work with, easy to implement, and surprisingly effective.

II. The Robot

The robot we use for this research is a statically stable mobile manipulator that our lab, the Healthcare Robotics Lab, assembled in early 2009 (see Figure 2). It consists of arms from MEKA Robotics (MEKA A1), an omni-directional mobile base from Segway (RMP 50 Omni), and a

Fig. 3. Examples of a human using his hand as a hook, and corresponding orientations for the robot's hook end effector.

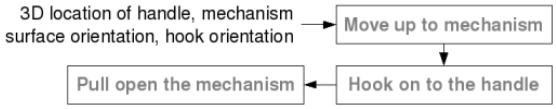

Fig. 4. Figure showing the input to the system and the three main actions that the robot executes in this paper to pull open doors and drawers.

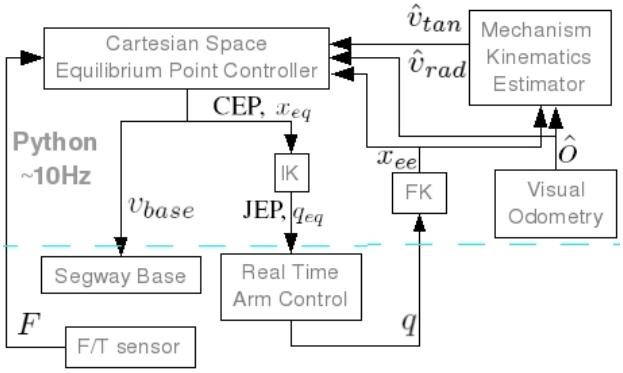

Fig. 5. Figure showing the overall control structure.

1-DoF linear actuator from Festo that can lift the manipulator and sensors from ground level to 1.2m above the ground. Distinctive features of this robot include the use of series elastic actuators [22] in all 14 DoF of the two arms (7 DoF each) and four Mecanum wheels for the base.

For this work, a hook serves as the end effector (Figure 3). We designed the hook, printed it with a 3D printer, and then applied rubber to its surfaces to increase friction. Section XI provides a link to the CAD design of the hook. One can think of this as a model of the human hand when a person uses a finger or fingers to hook around a handle and pull something open (Figure 3). We also took inspiration from prosthetic hooks, which are used with remarkable versatility and effectiveness. A hook has the advantage of being effective for a variety of handles, including recessed handles that would be difficult to grasp.

A. The Software and the Sensors

A Mac Mini running Ubuntu GNU/Linux performs all of the computation for sensing and high-level control. There is also a Dell Studio Hybrid that runs Ubuntu GNU/Linux with a kernel patched with RTAI for real-time operation. It performs computations for the MEKA arms. We have written all our software in Python and make use of a variety of open source packages including SciPy [23], KDL, ROBOOP, and ROS [24].

For this work, the robot primarily uses haptic and proprioceptive sensing. The robot senses forces and torques using a wrist-mounted 6-axis force/torque sensor (ATI Mini40 from ATI Industrial Automation). The arm's joints sense their joint angles and torques, but the current behaviors only use the joint torques implicitly in the context of virtual spring control.

1) The User Interface: For an assistive application, we have previously shown interfaces by which a person can designate a 3D location in the world using a laser pointer and a touch screen [25]. For this work, the robot uses a calibrated camera registered with a tilting laser range finder to present the user with a point-and-click interface with which the user can select the location of a handle to be manipulated. The user first uses a mouse to click on a handle in the image. The robot then estimates the 3D position associated with this clicked location using the registered 3D point cloud from the tilting laser range finder. It simply selects the 3D point that is closest to the ray associated with the user selected pixel. The user also specifies the direction in which the handle should be hooked.

III. System Description

Figure 4 shows the three main actions that the robot executes in this paper to pull open doors and drawers. We do not address the problem of finding a good place to hook onto a door or drawer in this work. Instead, we provide the robot with a 3D location to which the hook should be moved, the direction in which the hook should be oriented. The hook can be in one of two orientations, upward and to the left, as shown in Figure 3. In the experiments in this paper, the starting orientation of the robot is normal to the plane of the front of the door or drawer.

After the user provides the requisite information, the robot navigates to the mechanism and aligns itself parallel to the surface using visual odometry from a camera looking down at the floor. It then executes a behavior to hook onto the handle, explained in Section VIII-C. Finally, it executes the pulling behavior described in Section V to operate the mechanism.

A. Control Structure

Figure 5 shows an overall block diagram for the system. We describe the control blocks below:

1) Cartesian Equilibrium Point Controller: This is the main control block for the system. It is responsible for adjusting the Cartesian space equilibrium point (CEP) in the world frame. At each time step, it changes the CEP so as to operate the door or drawer and keep the hook on the handle. It then uses inverse kinematics (IK) to compute a joint space equilibrium point (JEP) corresponding to the CEP, and outputs the JEP to the real-time arm controllers that implement the virtual springs in the arm. When the CEP is at risk of falling outside of the arm's workspace, the controller commands the base to move the robot such that the CEP will move towards the interior of the arm's workspace. The inner workings of this block are described in more detail in Sections V and VI. The Cartesian Equilibrium Point Controller runs at approximately 10Hz.

2) Mechanism Kinematics Estimator: This block monitors the trajectory of the tip of the robot's hook in the world frame, and uses this information to infer the kinematics of the mechanism that the robot is operating. It assumes that the mechanism will either be a rotary joint or a prismatic joint whose trajectory lies in a plane parallel to the floor. It fits a circle to the Cartesian trajectory of the hook end effector to estimate the location of the axis of rotation [1].

The Mechanism Kinematics Estimator uses forward kinematics (x_{ee}) and visual odometry ($\hat{o}$) to estimate trajectory of the tip of the hook in the fixed world frame. It runs at approximately 5Hz.

3) Visual Odometry: The robot uses a floor-facing camera placed underneath the mobile base with a light ring to compute visual odometry. As it moves, the robot tracks visual features across sequential images of the floor. Due to wheel slip, we found odometry based on the wheel encoders to be poor. Visual odometry estimates the position and orientation of the mobile base, $\hat{o}[t]$, at approximately 10Hz.

4) Real Time Arm Control: This control block runs at 1kHz on the Dell Studio Hybrid, and simulates virtual viscoelastic springs for all joints of the manipulator except two wrist joints (J5 and J6 in Figure 2). This implementation of impedance control is detailed in Section IV.

5) Segway Base: The omni-directional mobile base accepts velocity commands, v_{base}, from the Cartesian equilibrium point controller.

6) F/T sensor: The force-torque sensor is at the base of the hook, on the wrist of the robot's arm and outputs 6-axis force/torque measurements.

IV. Simulating Viscoelastic Springs at the Manipulator Joints

On the Dell Studio Hybrid, a control loop simulates virtual viscoelastic springs for all joints of the manipulator except two wrist joints (J5 and J6 in Figure 2). At 1kHz, this control loop computes a torque vector $\tau = [\tau_0 ... \tau_4]$, which consists the torques applied to the three joints in the shoulder (J0, J1, J2), one joint in the elbow (J3), and the wrist roll joint (J4). τ is computed as the sum of two torque vectors:

$$\tau \quad = \quad -g(q) + (-K_p(q - q_{eq}) - K_d\dot{q}) \qquad (1)$$

The first torque vector, $g(q)$, is the torque due to gravity as a function of the current joint angles q. Subtracting it provides gravity compensation. The second torque vector, $-K_p(q-q_{eq})-K_d\dot{q}$, simulates a torsional, viscoelastic spring with constant stiffness and damping at each joint. K_p and K_d are diagonal stiffness and damping matrices, and q_{eq} is the joint space equilibrium point (JEP).

For the wrist joints J5 and J6, the robot uses position control that relates the motor output to joint encoder readings and ignores torque estimates from the deflection of the SEA springs. Consequently joints J5 and J6 are held stiff, except for the passive compliance of the SEA springs and cables connecting the SEAs to the joints.

For a detailed block diagram of the control structure for the arm, please refer to our previous work [1].

V. An Equilibrium Point Controller for Pulling Open Doors and Drawers

In this section, we describe a controller that generates Cartesian equilibrium point (CEP) trajectories in a fixed world frame that enable the robot to pull open novel doors and drawers. The fixed world frame, denoted with the superscript w, is attached to the initial location of the handle. The X axis of this coordinate frame is normal to the surface of the mechanism and points away from the robot, and the Z axis points vertically up. This is in contrast to our previous work in which the CEP was specified with respect to a frame of reference attached to the robot's torso. By specifying the CEP with respect to the world frame, we can consistently use it even when the robot moves its base.

This controller uses 6-axis force feedback from the wrist-mounted force/torque sensor (F), estimates of the mechanism kinematics, the end effector position (x_{ee}), and the translation and rotation of the omni-directional base ($\hat{o}$), see Figure 5.

At each time step, t, the controller computes the CEP, $x^w_{eq}[t]$, by adding a vector intended to operate the mechanism, $m^w[t]$, and a vector intended to keep the hook from slipping off of the handle, $h^w[t]$, to the previous CEP, $x^w_{eq}[t-1]$. So,

$$x^w_{eq}[t] \quad = \quad x^w_{eq}[t-1] + m^w[t] + h^w[t] \qquad (2)$$

where the superscript w denotes that all the coordinates are in the fixed world frame. Section V-A details how the vectors $m^w[t]$ and $h^w[t]$ are computed.

$x^w_{eq}[t]$ is then transformed into an equilibrium point in the local coordinate frame of the torso, $x^l_{eq}[t]$, using

$$x^l_{eq}[t] \quad = \quad T^l_w(\hat{o}[t-1]) \cdot x^w_{eq}[t] \qquad (3)$$

where T^l_w is the matrix that transforms points from the world coordination frame to the local coordinate frame of the torso. T^l_w is parameterized by the translation and rotation of the omni-directional base estimated by visual odometry, $\hat{o}[t-1]$.

Finally, the controller computes the joint space equilibrium point (JEP), $q_{eq}[t]$, from $x^l_{eq}[t]$ as

$$q_{eq}[t] \quad = \quad \text{Inverse Kinematics}(x^l_{eq}[t]) \qquad (4)$$

using the inverse kinematics (IK) solver from KDL[1]. When computing q_{eq}, the IK solver is seeded with the previous JEP in the trajectory. If no previous JEP exists, the robot uses a look-up-table to find a configuration of the arm with which to seed the IK solver.

A. Operating the Mechanism While Staying Hooked Onto the Handle

Based on the estimate of the location in the XY plane of the axis of rotation of the mechanism, the controller defines tangential and radial unit vectors $(\hat{v}^w_{tan}[t], \hat{v}^w_{rad}[t])$ for the trajectory of the handle. This is comparable to estimating a task frame [2], [26], [27]. Starting at $t = 0$, until the mechanism kinematics estimator has enough points

[1]Kinematics and Dynamics Library (http://www.orocos.org/kdl)

to estimate the kinematics, the controller moves the CEP in a linear trajectory towards the robot.

Using the radial and tangential unit vectors, the robot factors the force measured by the wrist force-torque sensor into estimated tangential and radial components, $(\hat{F}_{tan}[t], \hat{F}_{rad}[t])$.

$m^w[t]$ (see Equation 2), a vector of constant magnitude oriented in the direction of the estimated motion tangent, is calculated as

$$m^w[t] \quad = \quad 1cm \cdot \hat{v}^w_{tan}[t] \tag{5}$$

By itself, $m^w[t]$ would tend to create a CEP trajectory that looks similar to the trajectory traced out by the handle of the mechanism. To prevent the hook from slipping off the handle, the controller computes $h^w[t]$ (see Equation 2) as

$$h^w[t] \quad = \quad 0.1cm/N \cdot (\hat{F}_{rad}[t] - 5N) \cdot \hat{v}^w_{rad}[t] \tag{6}$$

Hence, $h^w[t]$ is a vector, parallel to the radial unit vector $(\hat{v}^w_{rad}[t])$, whose length is determined by a proportional controller that attempts to keep the radial force applied to the handle by the hook at 5N.

In addition, the robot tries to keep the orientation of the hook aligned with the estimated task frame subject to the joint limits of the arm. The goal of keeping the hook aligned with the task frame is to reduce the chance of the hook slipping off while operating the mechanism.

B. Stop Conditions

While updating q_{eq}, the robot looks for three types of stop conditions. First, if the magnitude of the force measured by the wrist force-torque sensor exceeds a maximum force threshold $F_{th}[t]$, the robot stops. While operating the mechanism, the robot computes $F_{th}[t]$ as

$$F_{th}[t] \quad = \quad \begin{cases} 80N & t \leq t_{mv} \\ min(\|F[t_{mv}]\| + 30N, 80N) & \text{otherwise} \end{cases} \tag{7}$$

$$t_{mv} \quad = \quad min\{t \text{ s.t. } \|x^w_{ee}[t] - x^w_{ee}[0]\| \geq 10cm\} \tag{8}$$

where t_{mv} is the time when the tip of the hook end effector has moved by a distance greater than 10cm, and $F[t_{mv}]$ is the measured force at time t_{mv}. $F_{th}[t]$ is initialized to 80N, and is adapted to be 30N greater than $F[t_{mv}]$ after time t_{mv}. This addresses the higher initial forces that are often required when opening doors and drawers.

Second, if the magnitude of the force drops below 1N for more than one second, the robot assumes that its hook end effector has slipped off the handle and stops. Finally, if the CEP trajectory leaves the workspace of the arm, the robot stops. Under some circumstances, the CEP trajectory could be allowed to leave the workspace, but we do not address these situations in this work.

VI. A Controller for Moving the Omni-Directional Base

The goal of this controller is to keep the Cartesian equilibrium point (CEP) and the tip of the robot's hook in the interior of the arm's workspace.

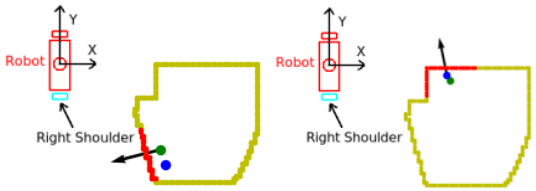

Fig. 6. Figure illustrating the computed direction of motion for the omni-directional base (black arrow) to move the equilibrium point (green) or the tip of the hook (blue) away from the boundary of the planar workspace of the arm (yellow). The red points are those that are used to compute the direction of motion for the mobile base, as detailed in Section VI.

Let $Bndry$ be the set of points on the boundary of the planar Cartesian workspace of the arm. The controller first computes

$$x^l_{close} \quad = \quad \underset{x \ \epsilon \ \{x^l_{eq}, x^l_{ee}\}}{argmin} \ (distance(x, Bndry)) \tag{9}$$

where x^l_{eq} is the CEP, x^l_{ee} denotes the coordinates of the robot's end effector, and the superscript l denotes that all the coordinates are in the local frame of the torso.

To determine the direction in which to move the omni-directional base, we use a reactive control technique that has been used for controlling mobile robot bases [28]. We compute the sum of "repulsive forces", r, on x^l_{close} as

$$r \quad = \quad \sum_{p \epsilon C} \frac{(x^l_{close} - p)}{\|x^l_{close} - p\|^2} \tag{10}$$

$$C \quad = \quad \{p \ \epsilon \ Bndry \ s.t. \ \|x^l_{close} - p\| < 10cm\} \tag{11}$$

where C is the set of points in $Bndry$ that are less than $10cm$ from x^l_{close}.

The controller commands the base to move at a speed of 15cm/s opposite to the resultant vector (r) by setting the commanded base velocity (v_{base}) to

$$v_{base} \quad = \quad -\frac{r}{\|r\|} \cdot 15cm/s \tag{12}$$

If x^l_{close} is greater than 5cm from $Bndry$, the controller sets $v_{base} = 0$. Figure 6 shows the computed base motion direction for two cases: $x^l_{close} = x^l_{eq}$ and $x^l_{close} = x^l_{ee}$.

VII. Coordinating the Omni-Directional Base and the Compliant Arm

Equation 2 updates the Cartesian equilibrium point (CEP) trajectory in a coordinate frame attached to the initial location of the handle and fixed in the world. Equation 3 then transforms the CEP into the local frame of the torso using the position and orientation of the mobile base, thereby coordinating the motion of the mobile base and the arm.

If the distance of x^l_{close} (Equation 9) from the boundary of the planar workspace of the arm falls below 3cm, we modify Equation 2 to be

$$x^w_{eq}[t] \quad = \quad x^w_{eq}[t-1] + h^w[t] \tag{13}$$

i.e. we set $m^w[t]$, the vector intended to operate the mechanism, to zero if x^l_{close} is near the workspace boundary. We now allow for the motion of the base to move the CEP and the tip of the hook away from the workspace boundary.

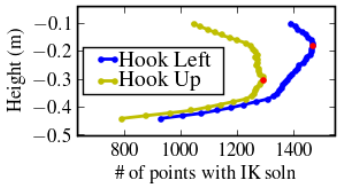

Fig. 7. This figure shows the workspace in planes parallel to the ground, estimated as the number of points in a grid of 1cm resolution that have an inverse kinematics solution. The red dots denote the heights with the maximum number of points (-0.3m and -0.18m).

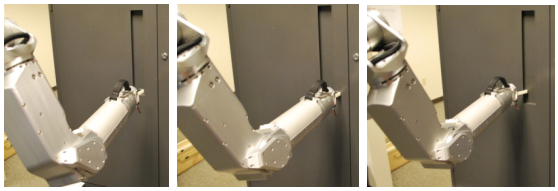

Fig. 8. Figure illustrating the two compliant motions that the robot executes to try to obtain a firm hooking grasp. **Left:** Starting position of the hook. **Middle:** Motion towards the mechanism until contact with the surface. **Right:** Lateral motion along the surface while pushing against it.

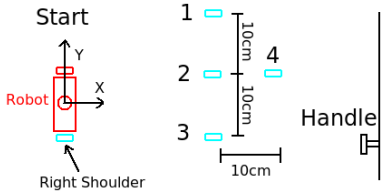

Fig. 9. This figure shows the approximate starting position of the robot and the four different target positions of the robot relative to the selected location of the handle after navigating to the handle and prior to the start of the manipulation behavior for the experiments of Section IX. Ideally, if the robot navigated precisely to target position 2, the selected handle location would be in the middle of the arm's planar workspace. The remaining three target positions are defined relative to target position 2.

VIII. Sources of Variation

The pulling controller depends on several factors: the way in which the handle has been hooked, the stiffness settings of the joints, the initial posture of the arm, and the pose of the body relative to the handle. Within this section, we describe how the robot selects these parameters.

A. The Pose of the Body Relative to the Handle

During autonomous activities, the position of the base and torso relative to the handle will be likely to vary due to uncertainties (e.g., perception), limited precision (e.g., motion of the base), task constraints (e.g., obstacles), and other challenges that accompany real-world operation. We wish to verify empirically that our proposed controllers are robust to these forms of variation.

1) Height of the Torso: Qualitatively, we found that system performance was not sensitive to the height of the torso relative to the handle. Consequently, we chose to fix the height of the torso relative to the handle for a given hook orientation. We chose this height by searching for a value that would maximize the area of the planar Cartesian workspace of the end effector. We estimated the area of the workspace by using the IK solver to sample over achievable end effector positions as shown in Figure 7. The robot attempts to keep the height of the torso relative to the handle as close to the height with the maximum estimated area as possible, subject to the joint limits of the linear actuator (Figure 2).

B. The Initial Posture of the Arm

Given the fixed pose of the wrist, the height of the torso, and the planar pose of the base, the arm must reach the handle such that it is hooking it. This leaves one DoF remaining in the 7DoF arm. We initialize the posture of the arm such that the plane formed by the shoulder, elbow and wrist is almost vertical with the elbow tilting slightly away from the torso.

C. A Behavior for Hooking the Handle

The robot updates its estimate of the location of the handle as it navigates close to it. Once the handle is within the workspace of the arm, it attempts to firmly hook the handle before executing the pull behavior.

It does this through two compliant motions. First, it moves the arm with a linear CEP trajectory towards the mechanism until it detects contact with the surface of mechanism using the wrist force-torque sensor at a point on the surface offset from the handle location. The robot then moves the hook laterally toward the handle along the surface of the mechanism. It uses a bang-bang controller that at each time step moves the CEP out of the surface by 0.2cm if the pushing force exceeds 3N, and into the door by 0.2cm if the force falls below 1N. Pushing against the surface allows the robot to hook onto recessed handles.

The robot continues the lateral motion along the surface until a force threshold indicates contact, or the hook has moved a distance of 10cm. These two motions are illustrated in Figure 8.

D. Selecting the Stiffness Values

For all of the experiments, we used the same stiffnesses for the five joints. The stiffness gains for the virtual viscoelastic springs for J0, J1, and J2 at the shoulder were 20, 50, and 15Nm/rad, respectively. For J3 at the elbow the stiffness gain was 25Nm/rad, and for J4 at the wrist it was 2.5Nm/rad (see Figure 2). These stiffness settings are the same as those used in our previous work on pulling open doors and drawers [1]. Qualitatively, we have found that the robot's performance is insensitive to changes in the relative stiffness [1].

IX. Evaluation

We evaluated the performance of the robot on 10 different mechanisms: four cabinet doors that open to the right, three cabinet doors that open to the left, and three drawers.

We performed four trials for each mechanism. The robot started around 1m from the location of the handle, aligned with the surface of the mechanism. The task for the robot was to navigate up to the mechanism and operate it. We deemed a trial to be successful if the robot navigated to

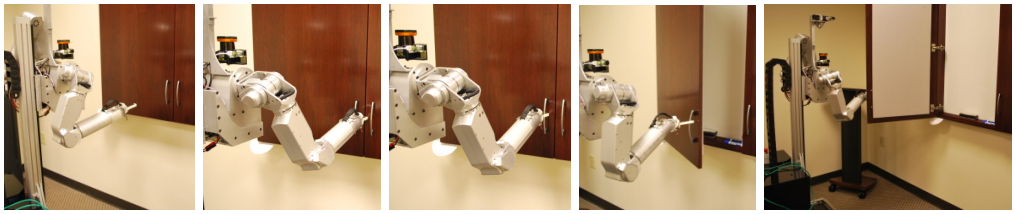

Fig. 10. Sequence of images that show the robot opening a door. The robot moves up to the mechanism, reaches out with the arm to make contact with the surface, moves the hook along the surface to hook onto the handle and then pulls open the door, moving the omni-directional base to increase the workspace of the robot.

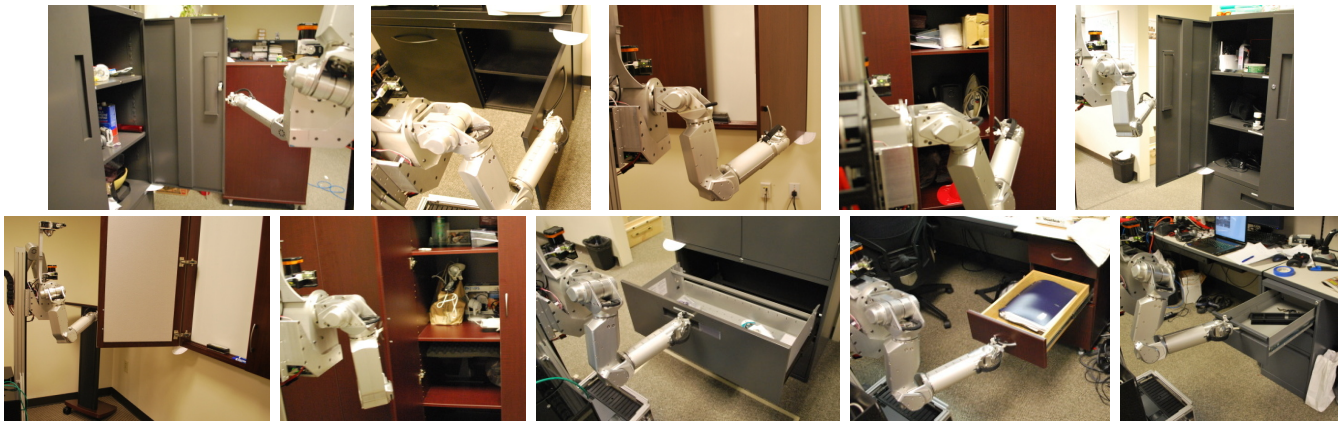

Fig. 11. Images showing the robot after it has operated the 10 mechanisms used in the experiments of Section IX. The order of images is consistent with the results of Table I.

the mechanism and opened it through an angle greater than 60° for rotary mechanisms or 30cm for linear mechanisms. We measured the angle using a protractor that we stuck to the mechanism which can be seen in the images in Figure 11. The input to the robot was the 3D location of the handle and an orientation for the hook.

To verify that the manipulation behavior is robust to the position of the robot, the approach behavior moved the robot to a different target position in each of the four trials. Figure 9 shows the four different target positions of the robot relative to the handle after navigating to the handle and prior to the start of the manipulation behavior. We varied the relative target position of the robot by 20cm parallel to the surface of the mechanism and 10cm normal to the mechanism. Ideally, if the robot navigated precisely to target position 2 in Figure 9, the selected handle location would be in the middle of the arm's planar workspace. The remaining three target positions were defined relative to target position 2. We began each of the four trials on one mechanism with the robot in approximately the same starting position.

Figure 11 shows the robot after it has pulled open each of the 10 mechanisms in one of the trials and Table I summarizes the performance of the robot. The order of mechanisms in Figure 11 and Table I correspond. The robot successfully opened rotary mechanisms through an angle greater than 60° in 26 out of 28 trials and pulled the linear mechanism through a distance greater than 30cm in 11 out of 12 trials. Overall, on 10 different mechanisms, the robot was successful in 37 out of 40 trials.

Two of the three failures were due to the robot not being able to hook onto the handle. We believe that this was due to a combination of errors in the odometry and the initial estimate of the location of the handle. The third failure occurred on Right Door 2, which has a large handle along which the hook can slip easily. The robot opened Right Door 2 through an angle of 40° and the pulling behavior stopped due to the force threshold being exceeded. In addition to this task failure, the hook slipping along the handle of Right Door 2 resulted in large errors in the robot's estimate of the kinematics for this mechanism, see Table I.

X. Discussion and Conclusion

In this paper we have shown that equilibrium point control (EPC) can enable a robot to autonomously approach and open doors and drawers for which only the location and the orientation of the handle have been provided. Through 40 trials with 10 different doors and drawers, we have empirically demonstrated that the robot can robustly operate different mechanisms and infer their kinematics without detailed prior models.

Further, we have shown that EPC can be used to coordinate the motion of the robot's omni-directional mobile base and compliant arms. This results in improved performance over our previous work [1] and other research that holds the base stationary while the arm manipulates the world. The robot's performance is more robust to the size of the mechanisms, and the position of the robot relative to the mechanism when it grasps the handle. Allowing the base to move during

TABLE I

PERFORMANCE OF THE ROBOT ON 10 MECHANISMS.

Mechanism	Angle/Distance pulled		Estimated Radius		Measured Radius	Max Magnitude of Total Resultant Force		Success Rate
	mean	std	mean	std		mean	std	
Right Door 1	85.0°	11.2°	0.37m	0.01m	0.39m	25.8N	3.6N	4/4
Right Door 2	67.5°	17.9°	1.77m	1.63m	0.17m-0.44m	33.5N	4.7N	3/4
Right Door 3	75.0°	3.5°	0.58m	0.03m	0.57m	19.7N	1.1N	4/4
Right Door 4	85.0°	8.7°	0.4m	0.04m	0.42m	23.9N	3.0N	4/4
Left Door 1	83.8°	8.2°	0.33m	0.02m	0.34m	34.3N	3.0N	4/4
Left Door 2	103.8°	8.2°	0.47m	0.02m	0.57m	30.9N	2.9N	4/4
Left Door 3	93.3°	2.4°	0.33m	0.05m	0.41m	34.7N	3.2N	3/4
Drawer 1	0.44m	0.0m				38.4N	2.0N	4/4
Drawer 2	0.48m	0.0m				37.1N	4.3N	3/4
Drawer 3	0.39m	0.0m				39.6N	3.1N	4/4

manipulation increases the effective workspace of the robot, enabling it to open doors over greater angular ranges and drawers over greater distances.

A number of challenges remain before robots start operating these mechanisms in unstructured environments. These include operating mechanisms that require large forces to open (e.g. spring loaded doors), and opening doors and drawers in constrained places. Further, robots should be able to detect and handle unexpected collisions between the arm, the mobile base, the mechanism, and the environment. We hope to address these challenges in future work. We expect the low mechanical impedance for contact at any point along our robot's arms to be useful in addressing these challenges.

XI. SUPPLEMENTARY MATERIAL

Supplementary material including the Python code, video and a CAD model of the hook is available at:
`www.hsi.gatech.edu/hrl/epc-icra10.shtml`

XII. ACKNOWLEDGEMENTS

We thank Chad Jenkins and Mike Stilman for helpful discussions during this research. We thank Marc Killpack and Travis Deyle for their work on visual odometry. We gratefully acknowledge support from Willow Garage and NSF grant IIS-0705130.

REFERENCES

[1] A. Jain and C. C. Kemp, "Pulling Open Novel Doors and Drawers with Equilibrium Point Control," in *Humanoids*, 2009.

[2] M. Prats, S. Wieland, T. Asfour, A. del Pobil, and R. Dillmann, "Compliant interaction in household environments by the Armar-III humanoid robot," in *Humanoids*, 2008.

[3] R. Diankov, S. Srinivasa, D. Ferguson, and J. Kuffner, "Manipulation Planning with Caging Grasps," in *Humanoids*, 2008.

[4] G. Pratt, "Low impedance walking robots 1," *Integrative and Comparative Biology*, vol. 42, no. 1, pp. 174–181, 2002.

[5] S. Buerger, "Stable, High-Force, Low-Impedance Robotic Actuators for Human-Interactive Machines," Ph.D. dissertation, MIT, 2006.

[6] M. Zinn, O. Khatib, B. Roth, and J. Salisbury, "Playing it safe [human-friendly robots]," *IEEE Robotics & Automation Magazine*, 2004.

[7] A. Jain and C. C. Kemp, "Behavior-based door opening with equilibrium point control," in *RSS Workshop: Mobile Manipulation in Human Environments*, 2009.

[8] C. Ott, B. Baeuml, C. Borst, and G. Hirzinger, "Autonomous opening of a door with a mobile manipulator: A case study," *IFAC Symposium on Intelligent Autonomous Vehicles*, 2007.

[9] H. Gomi and M. Kawato, "Equilibrium-point control hypothesis examined by measured arm stiffness during multijoint movement," *Science*, vol. 272, no. 5258, p. 117, 1996.

[10] F. Mussa-Ivaldi, N. Hogan, and E. Bizzi, "Neural, mechanical, and geometric factors subserving arm posture in humans," *Journal of Neuroscience*, vol. 5, no. 10, pp. 2732–2743, 1985.

[11] R. Shadmehr, "Equilibrium point hypothesis. In (Ed.), M. AA, editor, Handbook of brain theory and neural networks," 2002.

[12] N. Hogan, "The mechanics of multi-joint posture and movement control," *Biological cybernetics*, vol. 52, no. 5, pp. 315–331, 1985.

[13] E. Bizzi, N. Hogan, F. Mussa-Ivaldi, and S. Giszter, "Does the nervous system use equilibrium-point control to guide single and multiple joint movements?" *Behavioral and brain sciences(Print)*, 1992.

[14] M. Hinder and T. Milner, "The case for an internal dynamics model versus equilibrium point control in human movement," *The Journal of Physiology*, vol. 549, no. 3, pp. 953–963, 2003.

[15] L. Sentis and O. Khatib, "Synthesis of whole-body behaviors through hierarchical control of behavioral primitives," *International Journal of Humanoid Robotics*, 2005.

[16] X. Gu and D. Ballard, "An equilibrium point based model unifying movement control in humanoids," in *RSS*, 2006.

[17] M. Williamson, "Postural primitives: Interactive behavior for a humanoid robot arm," in *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, 1996.

[18] S. Migliore, "The Role of Passive Joint Stiffness and Active Knee Control in Robotic Leg Swinging: Applications to Dynamic Walking," Ph.D. dissertation, Georgia Institute of Technology, 2009.

[19] M. Williamson, "Robot arm control exploiting natural dynamics," Ph.D. dissertation, Massachusetts Institute of Technology, 1999.

[20] Y. Mukaibo, S. Park, and T. Maeno, "Equilibrium Point Control of a Robot Arm with a Double Actuator Joint," *International Simposium on Robotics and Automation*, 2004.

[21] D. Clapa, E. Croft, and A. Hodgson, "Equilibrium point control of a 2-DOF manipulator," *Journal of Dynamic Systems, Measurement, and Control*, vol. 128, p. 134, 2006.

[22] G. Pratt and M. Williamson, "Series elastic actuators," in *IROS*, 1995.

[23] E. Jones, T. Oliphant, P. Peterson, *et al.*, "SciPy: Open source scientific tools for Python," *URL http://www. scipy. org*, 2001.

[24] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, R. W. Eric Berger, and A. Ng, "ROS: an open-source Robot Operating System," in *Open-Source Software workshop of (ICRA)*, 2009.

[25] Y. S. Choi, C. D. Anderson, J. D. Glass, and C. C. Kemp, "Laser pointers and a touch screen: Intuitive interfaces to an autonomous mobile robot for the motor impaired," in *ACM SIGACCESS*, 2008.

[26] M. Mason, "Compliance and force control for computer controlled manipulators," *IEEE Transactions on Systems, Man and Cybernetics*.

[27] H. Bruyninckx and J. De Schutter, "Specification of force-controlled actions in the "Task Frame Formalism": A Synthesis," *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 4, pp. 581–589, 1996.

[28] R. Arkin, "Motor schema-based mobile robot navigation," *International Journal of Robotics Research*, vol. 8, no. 4, pp. 92–112, 1989.